AD-A009 365

PRELIMINARY SPECIFICATIONS FOR AN AWG-9 WCS MAINTENANCE TRAINER FOR THE F-14 AIRCRAFT

Joseph W. Rigney, et al

University of Southern California

Prepared for:

Naval Training Equipment Center
Advanced Research Projects Agency

April 1975

TECHNICAL REPORT: NAVTRAEQUIPCEN 74-C-0065-1

PRELIMINARY SPECIFICATIONS FOR AN AWG-9 WCS MAINTENANCE TRAINER FOR THE F-14 AIRCRAFT

Joseph W. Rigney
Douglas M. Towne
D. Kirk Morrison

University of Southern California
Los Angeles, CA 90007

Contract N61339-74-C-0065
(213) 746-2127
Effective Date: 1 January 1974
Expiration Date: 1 January 1975
Contract Value: $75,000
NAVTRAEQUIPCEN Task No. 2752-03P01
Project Scientist: William J. King
(305) 646-5130

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 2310 Amend 1
Program Code No. 4W10

April 1975

DoD Distribution Statement

Approved for public release; distribution is unlimited.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the United States Government.

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	**READ INSTRUCTIONS BEFORE COMPLETING FORM**
1. REPORT NUMBER N61339-74-C-0065-1 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) PRELIMINARY SPECIFICATIONS FOR AN AWG-9 WCS MAINTENANCE TRAINER FOR THE F-14 AIRCRAFT	5. TYPE OF REPORT & PERIOD COVERED Final (1 Jan. - 31 Dec. 1974) 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joseph W. Rigney, Douglas M. Towne, D. Kirk Morrison	8. CONTRACT OR GRANT NUMBER(s) N61339-74-C-0065
9. PERFORMING ORGANIZATION NAME AND ADDRESS Behavioral Technology Laboratories University of Southern California Los Angeles, California 90007	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2752-03P01
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency Washington, D.C.	12. REPORT DATE April 1975 13. NUMBER OF PAGES 5
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Training Equipment Center Orlando, Florida	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Maintenance
CAI
Built-in-tests (BIT)
CAI System Design
Computer Graphics
Maintenance Trainer
Weapons control system
AN/AWG-9
F-14 aircraft

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report contains preliminary specifications for an AWG-9 WCS maintenance trainer. These were derived using a procedure described in earlier reports in this series which capitalizes on the hierarchical nature of serial action tasks to develop interactive segments of CAI for Naval technical training. In developing trainer specifications using this method, detailed task analytic and flowchart methods are utilized in conjunction with established principles of human learning

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

to construct a tentative instructional strategy. Implications of this strategy are then cast in terms of the realities of available hardware (e.g., computer graphics) and software (e.g., TASKTEACH) to determine preliminary training system specifications. Cost-benefit trade-offs are then made, and the entire process is iterated by successive approximation until a sufficiently specific guide to hardware acquisition and computer programming can be produced.

One of the more unique elements to the approach is that the natural interdependencies among serial action subtasks are allowed to influence very strongly the instructional strategy and computer monitoring of trainee performance, thereby greatly reducing the need for frame-by-frame construction of CAI materials.

SUMMARY

This report contains preliminary specifications for an AWG-9 WCS maintenance trainer. These were derived using a procedure described in an earlier report in this series. The specifications for a particular application of performance-structure-oriented CAI, such as this, follow from mapping specific features of task structures, contextual structures, and the instructional strategy to be used into the frameworks of general subsystems in CAI systems and of the major interactive loops in these systems. The resulting implications for hardware and software specifications then are developed by successive approximations until they are sufficiently specific to guide hardware acquisition and computer programming.

PREFACE

This study is another application of Computer Assisted Instruction (CAI) to Navy training problems funded by Advance Research Projects Agency (ARPA), performed by the University of Southern California, and monitored by the Naval Training Equipment Center. The Radar Intercept Officer's (RIO) job as noted in NAVTRAEQUIPCEN 71-C-0219-1, was the first technical area addressed in the manner described herein, and the AWG-9 Weapons Control System Maintenance Technician's job is the second. Both of these pace-setting studies were under the very able monitorship of Dr. Arthur Blaiwes of the Naval Training Equipment Center's Human Factors Laboratory.

The preliminary specifications described herein were designed to capitalize on such unique developments as the stand-alone computer graphics terminal, the power of minicomputers, and the dedication to a training philosophy based on the principle of student-initiated training scenarios. While these same developments were used to generate the RIO trainer earlier in this series, they have only begun to be used in teaching the complexities of avionics maintenance. In anticipation that principles learned in the operational world will transfer over to maintenance training, plans are afoot to develop a prototype trainer based on the specifications delineated in this report.

William J. King
WILLIAM J. KING, Ph.D.
Scientific Officer

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

The objectives of this research are to develop and to field-test a maintenance trainer, based on TASKTEACH program logic (Rigney, et al., 1972) and a simulated hands-on training, CAI testbed (Rigney, 1973). The trainer will be designed (1) to augment current system trainer facilities for training built-in-test (BIT) operators with an inexpensive, stand-alone CAI system, and (2) to give technicians an understanding of the test sequences in each BIT sequence in relation to the functional and physical organization of the AWG-9 Weapon Control System (WSC) in the F-14 aircraft.

As was pointed out in an earlier report in this series (Rigney, et al., 1974a), there are many reasons why operational equipment is ill-suited to teaching job skills. Weapon systems are becoming prohibitively expensive to use for training. Operational equipment lacks even the most elementary instructional features. Maintaining this equipment in peak operating states in schools is virtually impossible. This equipment cannot intentionally be put into malfunctioning states that would damage the equipment or be hazardous to the student. There is not enough of it to allow each student intensive practice in performing job skills.

Under these circumstances, it is reasonable to seek alternative methods of giving students troubleshooting training. In the approach used here, the man-machine interfaces involved will use interactive computer graphics to simulate "hands-on" operation. The general techniques that will be employed have been described (Rigney et al., 1974b). It also would be possible to employ full-scale front panel mock-ups on-line with the CAI testbed. Although this is not planned for the first version of the trainer, should the front panel hardware, controls, indicators, and panels become available at a later date, this alternative could be developed.

The principal advantages of this approach result from substituting a general-purpose digital computer, behind the man-machine interface, for the complex electronics of the AWG-9 WCS and the BIT circuitry. The fact that the actual circuitry is not behind this interface is, in fact, a distinct advantage, since all of the disadvantages of using the actual equipment for training are thereby avoided. The effects of any malfunction, however traumatic it might have been to the real equipment, can be described at the interface. There is no time lost in changing "bugged" weapon replaceable assemblies (WRA's), and no dependency on the necessarily very limited library of these to show the student what happens when the system fails. Beyond this, the powerful instructional features of a CAI system are available for automatically administering and controlling training for each student.

SECTION II

GENERATING SPECIFICATIONS FOR PERFORMANCE-STRUCTURE-ORIENTED CAI

A Procedure for Generating CAI System Specifications

Performance-Structure-Oriented (PSO) CAI was developed from consideration of the recent theoretical advances in cognitive psychology (Rigney, 1971) and from analysis of the structure of serial-action tasks (Rigney, et al., 1969). The PSO concept focuses on the serial-action characteristics of maintenance tasks. For example, certain functions must be performed before others, certain tasks must be performed simultaneously, or others may be carried out at any time. The key to the approach is that such interdependencies can be cast into a general format for programming purposes such that powerful computer monitoring and aiding of the instructional task can be realized. This conception of the organization of performance in terms of human information processing functions was first described in Rigney, et al., (1967). The emphasis in PSO CAI presently is on giving students the opportunity to learn essential subskills and to put them together into integrated performance. It is assumed that they will have been exposed to the necessary background knowledge by other delivery systems.

Procedures for generating specifications for PSO CAI were considered in an earlier report in this series (Rigney, 1974c). Some of the general concepts set forth there are reviewed here, as a matter of continuity. The "bases that must be touched" are illustrated in figure 1.

The analyst starts with available information about what is to be done and how, suggested by the arrows at the top of the diagram, and derives hardware and software specifications using procedures implied by the interconnected boxes in the diagram. When he is finished, he will have generated and organized these two categories of specifications for each of the necessary elements of a CAI system.

If it is a tactical or operational job, the "subject matter" might be air intercepts. The geometry of air intercepts would have to be analyzed, so that the air intercept environment could be simulated. If the job is maintenance of devices, then the structure of the device to be maintained must be analyzed. Generally, tactical jobs require quite different data bases and program logic than maintenance jobs. Task analysis in the diagram discerns the structure of serial-action tasks.

The instructional strategy is concerned with the content and sequencing of instruction. The serial sequence of information is the principal means for changing the behavior of the student. The instructional strategy will impact the composition of the instructional sequence: what items, problems, drills, external feedback, and other instructional aids are used, and in what order. These, in turn will require particular kinds of data structures and storage.

The scheduling of the instructional sequence will require, if done adaptively, a student monitor program. This program, in turn, must have access to a student performance history and, must contain or be able to

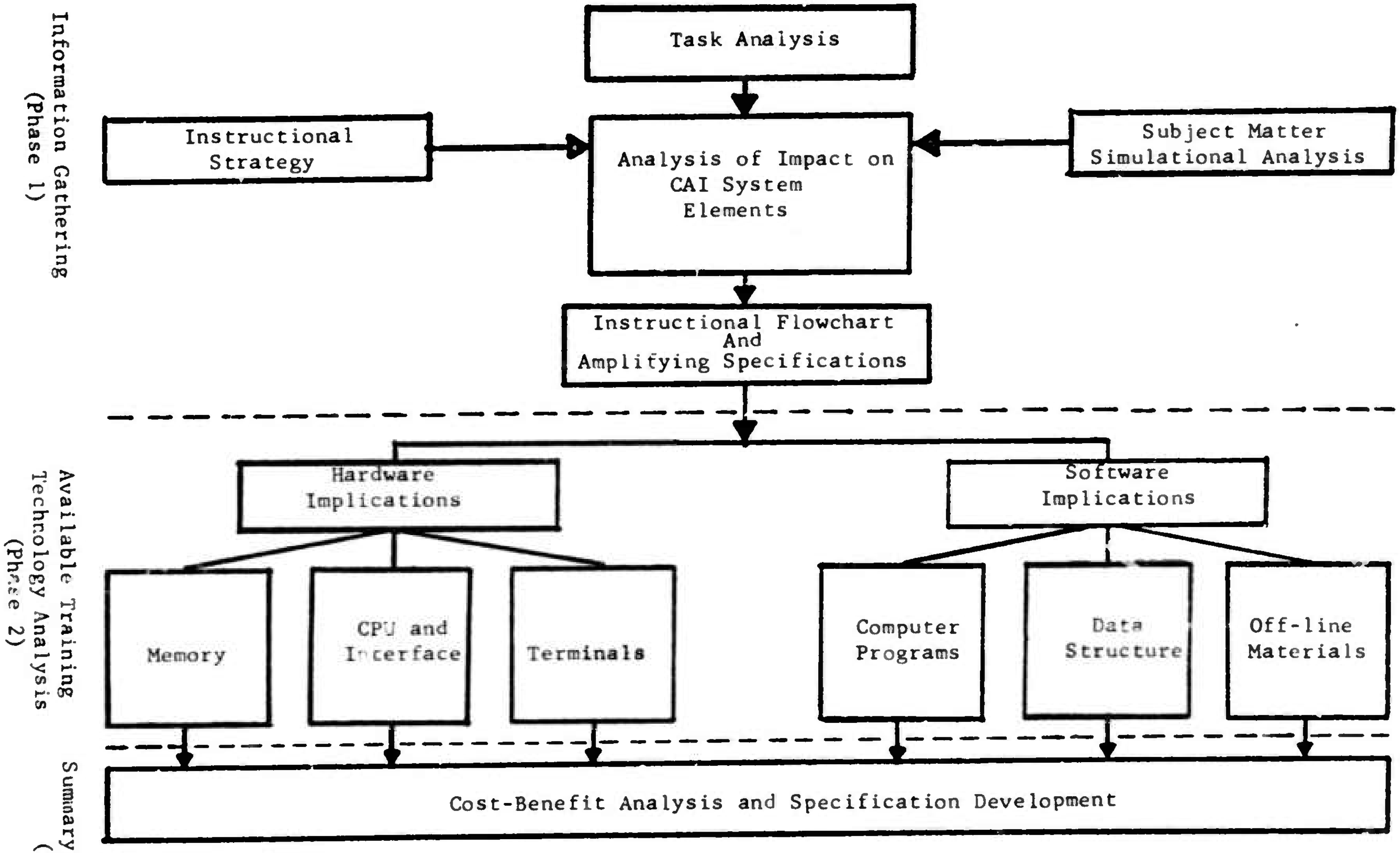


Figure 1. Outline of Procedure for Generating CAI System Specifications from an Analysis of the Performance to be Taught

call upon some type of optimizer, and/or sequence generator. (These CAI subsystems are illustrated in figure 2).

The impact analysis in figure 1 abstracts from task and subject matter analyses those features which contain information for the way CAI system elements are designed. Some examples illustrate this.

In the RIO trainer, the air intercept task structure requires real-time performance paced by a situational display showing fighter and bogey headings, speeds, turns, etc. This requirement was met by real-time animation using computer graphics programs. Also real-time animation requires a local processor and memory (in the terminal) because of high refresh rates needed.

As another example, for an AWG-9 maintenance trainer, an important feature of the subject matter is the built-in-test sequences that test different parts of the system. Each of eight BIT sequences is composed of a number of tests and subtests, which are performed automatically and in sequence, under program control in contrast to simpler devices where the technician selects the next test to perform. The implication of this is that there must be an executive routine in the trainer to sequence these tests, when the student is learning to use BIT.

In summary, the impact analysis looks at these tasks and contextual structures and identifies features that would impact on one or more of the six necessary CAI system elements. The implementation of these elements by hardware and/or by software then is determined and specifications for this implementation are generated in one or more of the six boxes at the bottom of the diagram. For example, program flowcharts are specifications for computer programs.

It is usually convenient to divide the development of specifications into phases. In Phase I, operations will be mostly concerned with information gathering and integration. It may be necessary to perform task, equipment, and tactical analyses as part of the development, or some of this information may be available in documentation. In either case, a subject matter expert is a practical necessity to select and to interpret documents and to fill in gaps in information.

A good way to integrate Phase I information is to construct an instructional flowchart. This will be an instructive exercise, because it forces planners to put down the surface structure of the student-program interaction in detail. The instructional flowchart will serve as the principal guide for Phase II operations, although it must be supplemented by other sources of information. Usually, several iterations are required before a flowchart can be produced at a sufficiently detailed level to be useful. Information supplementing the instructional flowchart would consist largely of the results of examining implications for each of the general CAI subsystems (see figure 2), in turn, of task and contextual structures and of the instructional decisions about how the operations required for each element will be implemented. During this exercise, the instructional flowchart will serve as a reminder of features that must be considered in more detail.

In Phase II, the objective is to describe the specific elements in the CAI system that will be designed to do the training. Given the outputs of Phase I: the instructional flowchart and the preliminary delineations of element operations, the analytical problem can be divided into hardware and software sections, and the effort can be concentrated on developing specifications in the last phase at the bottom of the diagram.

In summary, the whole process consists of relating specific applicational information: task and contextual structures, to general requirements of CAI system elements and to an instructional strategy to derive the specifications for specific implementations of CAI system elements that will do the job of training.

Of course, the analytical processes are not so clean and symmetrical as figure 1 implies. For one thing, the CAI hardware system usually has basic capabilities that would not change unless some unusual variable in the application took on values that exceeded some basic capability; e.g., for unusual interface devices such as front-panel mockups.

General CAI System Elements: CAI Subsystems

Any CAI system is composed of several subsystems, each performing a set of functions necessary to the total system performance. Functions of these subsystems have been described in earlier reports (Rigney, et al., 1973a, 1974c). Since the design of the AWG-9 WCS maintenance trainer is based on these general concepts, it is necessary to repeat a certain amount of these earlier descriptions, to achieve continuity in the sequence of thought expressed here. This overview is referenced to figure 2.

Student--Of course, just as is done for other elements in the CAI system, it is necessary to develop "specifications" for the student. Implicit assumptions about student characteristics must be made explicit. There always is some target population for whom the CAI system is to be designed. Characteristics of this population should be known in relation to prerequisites for the training. Adapting to individual differences among students is based on knowledge about student characteristics.

Rate of learning might be predicted from a mixture of achievement and intelligence tests. However, it is preferable to use adaptive control mechanisms to sense important individual differences and to adjust to them as the student is progressing, rather than to depend upon the classical psychometric approach, viz, batteries of tests and multiple regression, to try to predict what "treatment" should be assigned to each student.

Student-Program Interface -- This includes stimulus display, responses, and response records. Observe that responses include "trial responses" the student may take covertly before he commits himself to an observable response.

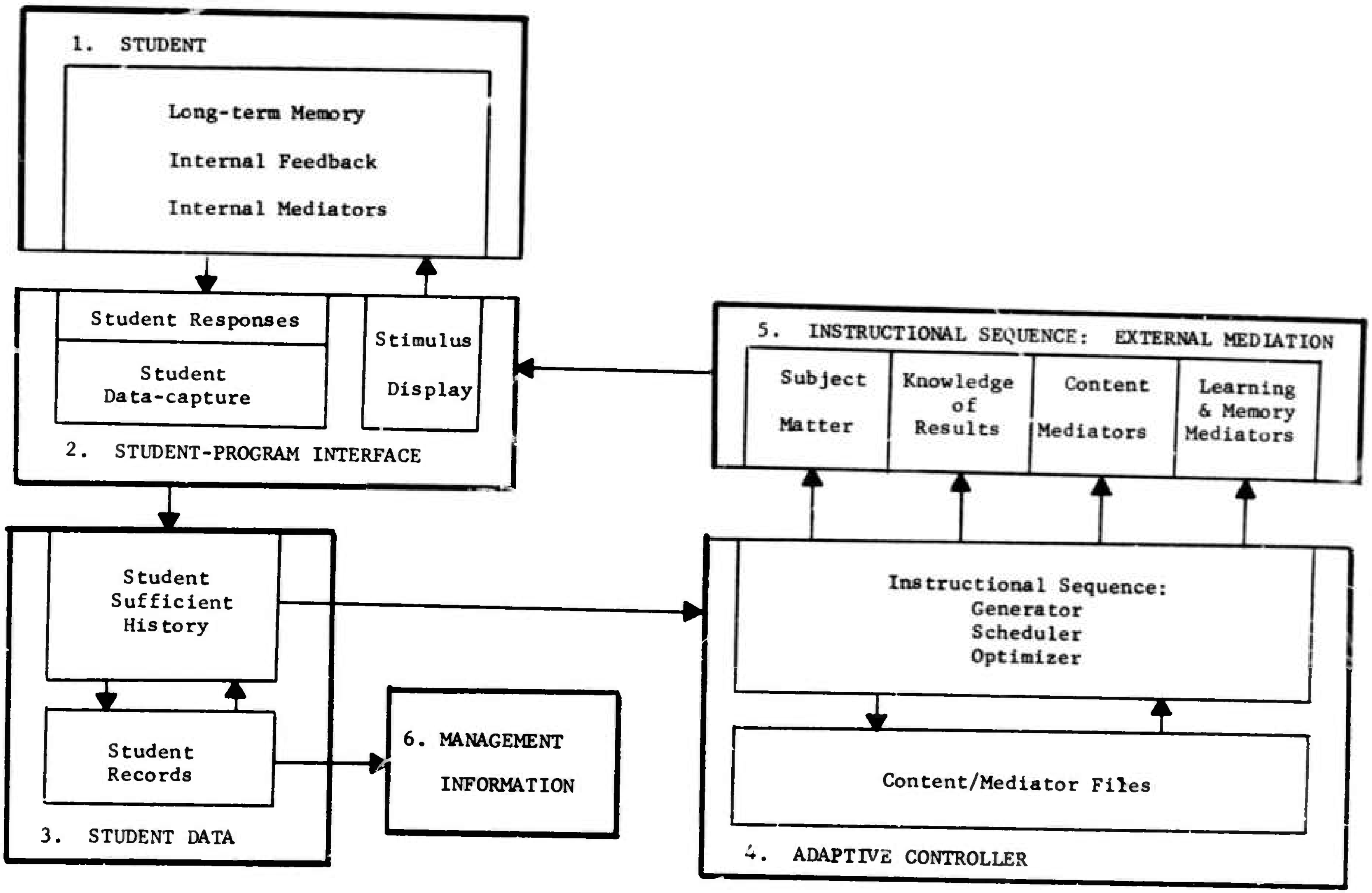


Figure 2. Outline of Major CAI Subsystems

Requirements here are to identify the stimulus displays, the response input devices, and the student data that must be captured. For the type of CAI system that is under development for the AWG-9 WCS, the man/machine interfaces in the job environment are simulated with interactive computer graphics. Methods for doing this are already well-developed at BTL (Rigney, et al., 1974b).

Student Data--This includes student performance history and student records. The latter would include processed response records plus other information about the student; e.g., intelligence test scores.

Adaptive Controller--This includes content/mediator files, essentially storage of course materials and auxiliary instructional materials, an instructional sequence optimizer, an instructional sequence scheduler, and an instructional content generator. These three instructional operators require some explanation. An optimizer would be some method for improving the effectiveness of the CAI system by optimizing learning rates under some set of constraints. Atkinson and Paulson (1972) have described general procedures for going about doing this.

The optimizer would identify an optimal instructional sequence for each student. The composition and sequencing of instruction is the principal way, if not the only way, to influence learning and retention in the CAI system. The optimizer would control an instructional sequence scheduler. Instruction must be sequenced by some mechanism and that mechanism must be described in the specifications for the CAI system.

The instructional content generator could be either of two types: software that "makes up" the instruction on the spot from simple descriptive data, as in the TASKTEACH programs, or a team of instructors who write programmed instruction frames, and generate the entire sequence ahead of time.

It is quite possible, as this laboratory has shown, for a computer to generate the instructional sequence for a troubleshooting problem step-by-step, while the problem is being solved by a student, and to do this for troubleshooting a wide variety of devices. It is possible as this laboratory also has demonstrated (Rigney, et al., 1972), to represent the structure of tasks in simple data-structures and to provide a general program to teach students how to perform a variety of tasks, either allowing them to compose their own individual instructional sequences or placing the instructional sequence under program control.

For the instructional content generator, the two major questions are: "What is the surface structure of the performance to be taught?" and "What are the content mediators that will be required in the training?" Content mediators are the operations and concepts in the material to be taught that bridge between student stimulus and student response. For example, in algebra, the mathematical operations and mathematical concepts required to solve a quadratic equation would be content mediators. This term is used to distinguish between these and learning and memory mediators, which are general operations the student may perform on broad categories of material to improve rate of learning and length of retention.

The content/mediator files in the adaptive controller store the material that is to be organized into an instructional sequence, or that is already organized into some "lesson" format. The way this material is stored and how much of it there is to store are important considerations. CAI requires a lot of storage, which usually takes the form of disc files. The random-access disc file is a common and almost indispensable feature of on-line systems. However, some CAI material may be stored more economically in peripheral devices in the form of slides or microfiches.

Instructional Sequence--As represented in the diagram, the instructional content would have been already composed by the generator and would be put into serial order by the scheduler, both in the adaptive controller. Thus, the instructional sequence is the output of operators in the adaptive controller.

This sequence contains knowledge of results, subject matter, and external mediators. It is the "input tape" to the student. It is necessary to distinguish between knowledge of results as something the system provides for the student, and internal feedback as something the central nervous system provides, using information that comes to it both from the external and the internal environments. Knowledge of results provides some information for internal feedback, but this information may or may not be useful. The knowledge of results provided by the system sometimes may be superfluous.

Since this material is at some intermediate difficulty level in a roughly hierarchical structure, there usually are a number of different concepts, relations, and skills to be learned. These must somehow be organized into a sequence and presented to the student's very limited input system in serial order. Thus, the fundamental problem is how to schedule the instructional sequence in a way that will cover the different concepts, rules, and operations to be taught, and still lead to optimum learning and retention rates.

Management Information--Student data are used to inform training managers about the progress of students and classes, and to relate this information to other variables that may be of interest to management. Extensive statistical analyses and tabulations may be performed, depending on management's requirements.

Feedback Loops--The CAI system contains feedback channels that must be used to learn about the student and to learn about the adequacy of the instructional sequence.

It is becoming increasingly clear, too, that the human nervous system also contains many feedback loops. These evidently are under central control (Powers, 1973). The implications of this mode of organization for the design of an instructional system are only beginning to be realized.

Student-CAI System Interactive Loops--CAI may be viewed as a series of complex interactions between and within the student and the CAI system, occuring moment-by-moment at the student-program interface (microinteractions), and extending over a longer period of time (macrointeractions). This conception is represented in figures 3 and 4.

Figure 3 is a highly generalized conception of these relationships. In figure 4, the two basic interactive loops have been superimposed on a simplified version of figure 2, whose elements were described above. Again, these figures have appeared in earlier reports.

Microinteractions are the fine-grained, moment-by-moment exchanges of information between the student and the CAI system. Macrointeractions are involved with scheduling instruction over longer periods of time. These interactions may be student-controlled, program-controlled,or mixed initiative. Currently, there is much controversy over which locus of control is better. In fact, some mixture of mixed-initiative control almost always is employed. The proportions that should be used in the mixture depend on many circumstances; maturity of the student, objectives of the training, the incentive system, relative emphasis on instruction versus practice, etc.

Each type of interaction becomes concerned with two kinds of structures: surface and deep. Surface structures are features of the interaction that are more or less visible at the student-program interface. Deep structures are processes which mediate the interactions and which reside with the student and in the CAI system logic--the computer programs.

Some of the kinds of surface and deep structure features that are considerations for planning micro and macrointeraction logic are suggested in figures 5 and 6. Obviously, very little is known about deep structures in the human nervous system that mediate learning and memory. But, at least, the interest of researchers has turned to these structures in recent years, and some of the resulting research does have exciting implications for instructional technology.

Deep structure considerations for programs include questions of CAI languages and their generality, of how much interactive logic to put in the programs and how much to distribute through the instructional sequence, and of how, in general, to utilize the great power and speed of the digital computer in education and training.

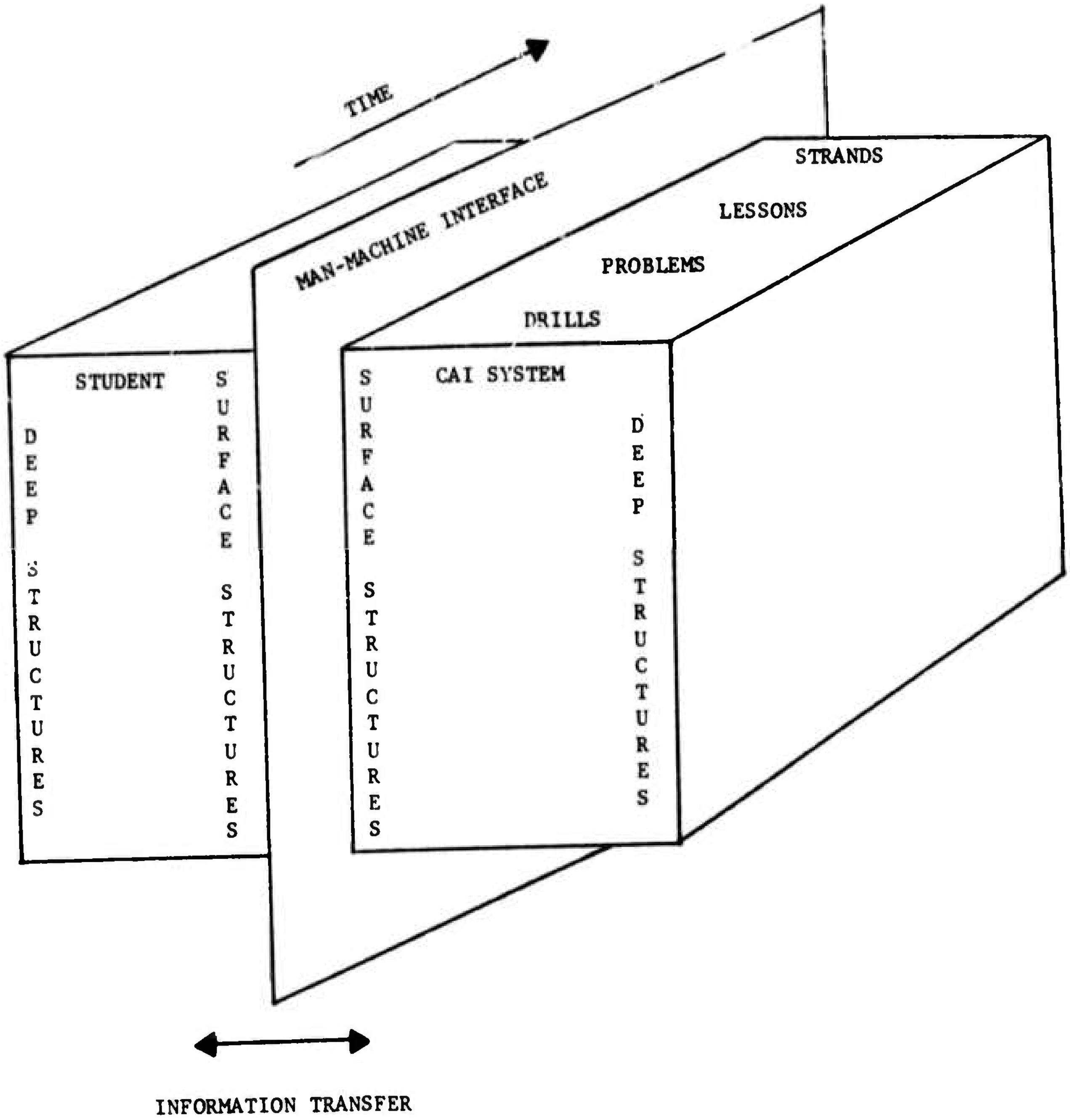


Figure 3. CAI as Interactive Communication Between Student and CAI System

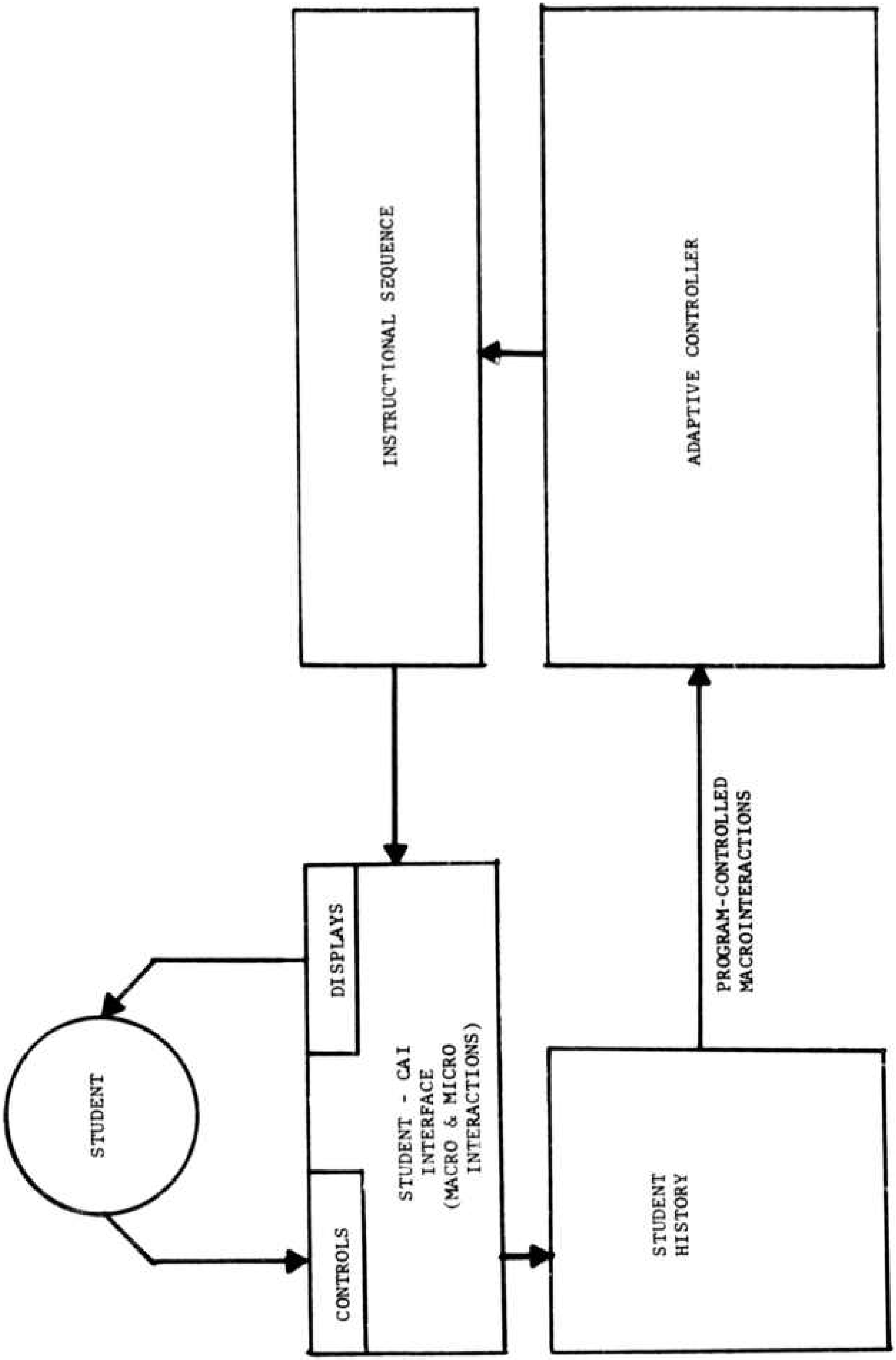

INSTRUCTIONAL SEQUENCE
ADAPTIVE CONTROLLER
STUDENT
DISPLAYS
STUDENT - CAI INTERFACE (MACRO & MICRO INTERACTIONS)
CONTROLS
STUDENT HISTORY
PROGRAM-CONTROLLED MACROINTERACTIONS

Figure 4. Micro and Macrointeraction Loops in CAI Systems

This overview of considerations for the design of CAI systems has been repeated here to provide all the information relating to the design of the AWG-9 WCS Maintenance Trainer in one place. Subsequent sections, devoted to this particular application, will take up these general elements in more detail, specifying how they will be implemented in this trainer. First, however, questions of training objectives will be considered.

	MICROINTERACTION	
	STUDENT	CAI SYSTEM
SURFACE STRUCTURES:	1. Observable performance	1. Information displays: Visual Auditory Touch 2. Input devices: Keyboards Light pens Touch panels Special
DEEP STRUCTURES:	1. Sensory-motor channels 2. Selective attention 3. Short-term memory 4. Short-term processing	1. Program logic to sustain micro-interactions 2. Suitable computer hardware to run program logic

Figure 5. Surface and Deep Structure Considerations for Microinteraction Loop

	MACROINTERACTION	
	STUDENT	CAI SYSTEM
SURFACE STRUCTURES:	1. Student histories (data)	1. Instructional sequence 2. System control commands
DEEP STRUCTURES:	1. Long term memory: Storage processes Relational structures Retrieval processes Retention intervals 2. Other organization and control processes in the CNS	1. Student data analysis routines 2. Adaptive control logic 3. Content and student data files 4. Lesson preparers or generators 5. Suitable hardware

Figure 6. Surface and Deep Structure Considerations for Macrointeraction Loop

SECTION III

SPECIFICATIONS FOR THE AWG-9 MAINTENANCE TRAINER

Sources of Training Objectives

Billet Qualification Standards--The Billet Qualification Standards for maintenance technicians for the F-14, revised 1 September 1973, contain objectives for formal academic training (NAMTRADET), practical training (Readiness Squadron) and on-the-job training (Operational Squadron). A minimum achievement level (MAL) is defined for each objective. The portions of the Billet Qualification Standards concerning the BIT sequences for the AWG-9 represent one type of official training objective. Additional revisions of these standards are likely in the future as the maintenance procedures and this weapon control system mature.

Top Level Flowcharts for Maintenance Procedures--The WCS testing and troubleshooting philosophy is described in flowchart form in figure 7 taken from NAVAIR 01-F14AAA-2-3-14. Again, this kind of formulation is subject to revision as more experience is accumulated with this aspect of F-14 maintenance. Original assumptions regarding the logistics of maintenance, which were one basis for this flowchart, are likely to be revised. The figure does, however, represent a procedural specification for the use of built-in-test sequences.

Logic Trees--Decision trees for maintenance actions are being produced by the hardware contractor. These are basically fault localization trees in which manual procedures, BIT sequences, and BEYOND BIT procedures, e.g., SPAM are all specified as appropriate maintenance actions at different points in the tree. These logic trees are quite detailed. At this point, they are in prepublication form, and are not available for inclusion in this report. Basically, however, they are straightforward fault-localization trees containing very detailed prescriptions of maintenance actions.

It is worth noting that such detailed prescriptions of actions for the technician to follow in doing fault localization are the current trend in dealing with the human factor in maintainability of complicated weapon systems.

Training Objectives

It is evident, from an examination of information from the above sources, that the maintenance technician should be thoroughly familiar with the BIT sequences, and with the kinds of tests and how they are made in each sequence. He also should be capable of operating each BIT sequence; from preinitialization actions, initialization actions, through accessing degraded

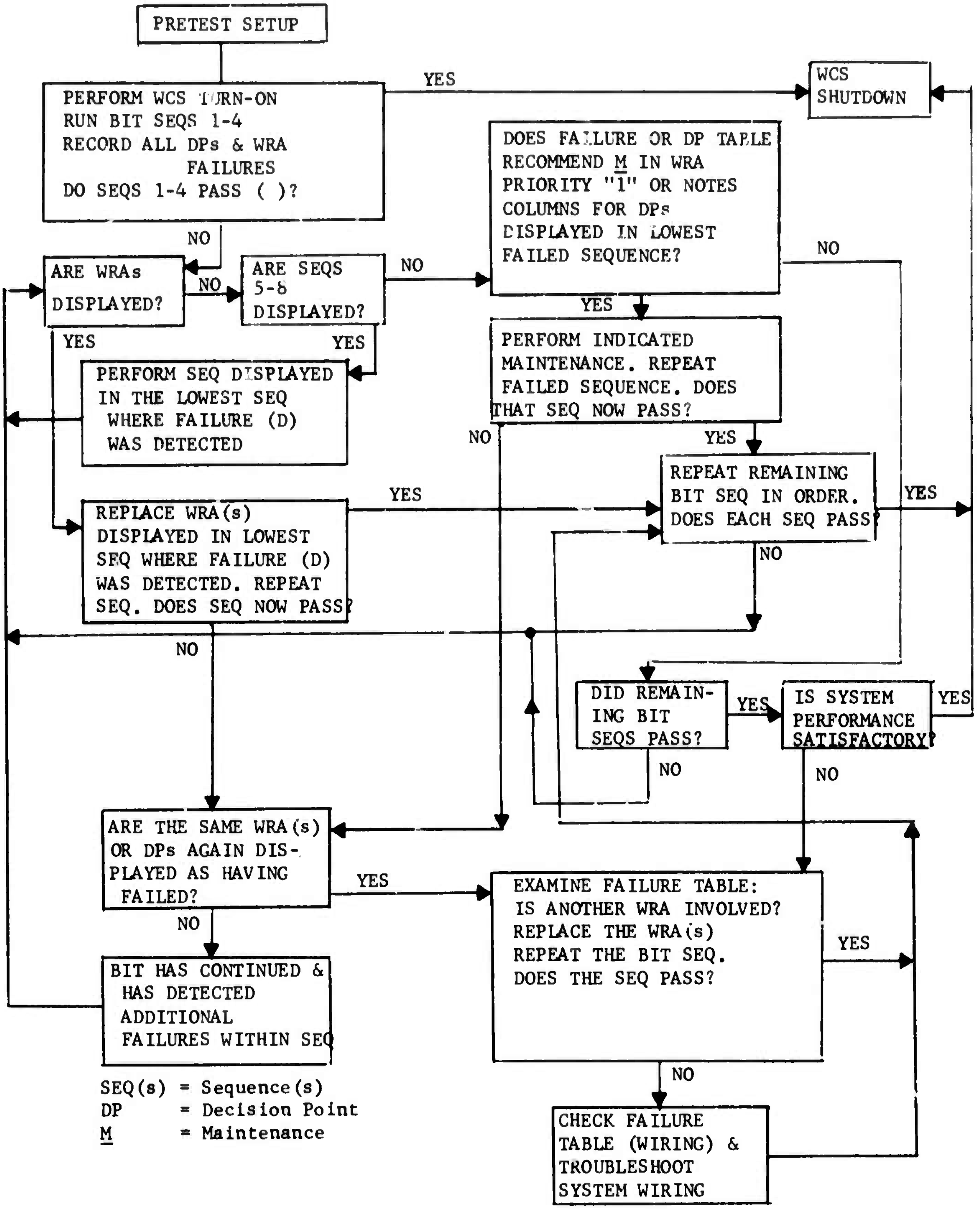

PRETEST SETUP
PERFORM WCS TURN-ON RUN BIT SEQS 1-4 RECORD ALL DPs & WRA FAILURES DO SEQS 1-4 PASS ( )?
YES
WCS SHUTDOWN
NO
ARE WRAs DISPLAYED?
NO
ARE SEQS 5-8 DISPLAYED?
NO
DOES FAILURE OR DP TABLE RECOMMEND M IN WRA PRIORITY "1" OR NOTES COLUMNS FOR DPs DISPLAYED IN LOWEST FAILED SEQUENCE?
NO
YES
YES
YES
PERFORM SEQ DISPLAYED IN THE LOWEST SEQ WHERE FAILURE (D) WAS DETECTED
PERFORM INDICATED MAINTENANCE. REPEAT FAILED SEQUENCE. DOES THAT SEQ NOW PASS?
NO
YES
REPEAT REMAINING BIT SEQ IN ORDER. DOES EACH SEQ PASS?
YES
YES
REPLACE WRA(s) DISPLAYED IN LOWEST SEQ WHERE FAILURE (D) WAS DETECTED. REPEAT SEQ. DOES SEQ NOW PASS?
NO
NO
DID REMAINING BIT SEQS PASS?
YES
IS SYSTEM PERFORMANCE SATISFACTORY?
YES
NO
NO
ARE THE SAME WRA(s) OR DPs AGAIN DISPLAYED AS HAVING FAILED?
YES
EXAMINE FAILURE TABLE: IS ANOTHER WRA INVOLVED? REPLACE THE WRA(s) REPEAT THE BIT SEQ. DOES THE SEQ PASS?
YES
NO
BIT HAS CONTINUED & HAS DETECTED ADDITIONAL FAILURES WITHIN SEQ
NO
CHECK FAILURE TABLE (WIRING) & TROUBLESHOOT SYSTEM WIRING
SEQ(s) = Sequence(s)
DP = Decision Point
M = Maintenance

Figure 7. WCS Testing and Troubleshooting Philosophy

mode assessment and fault isolation displays, to interpretation of displayed information. Finally, he should be similarly familiar with digital entry procedures, e.g. SPAM, and be capable of initiaiting SPAM and using the information it displays in the context of fault isolation trees. In this regard, the objectives in the Billet Qualification Standards, and in the other documents mentioned above, will be used.

"He" in the above paragraph actually is two technician levels, a BIT operator, and a more intensively trained senior technician. The latter might be more likely to use SPAM or FLYCATCHER, as BEYOND BIT procedures for isolating faults to smaller fault areas than one weapon replacable assembly (WRA), or for isolating the faulty WRA when BIT results require supplementary information.

Since PSO CAI is designed to give students practice in performing job tasks, the principal means for identifying when students have achieved training objectives will be time and error measures made on selected performance variables, and the principal means for assuring transfer to on-the-job performance will be simulation of the essential elements of job tasks, AWG-9 man-machine interfaces, and BIT functions. In short, PSO CAI simulates the job situation and assists the student in learning to perform job tasks until he is able to proceed on his own to meet performance standards.

Subject Matter Analysis

As implied in figure 1, all training problems have a subject matter, and the latter in this case is the use and general understanding of the BIT sequences in the AWG-9. The requirement for the analysis in this case is to identify those features of the weapon system that must be described, without going into additional detail. Analytical techniques developed for TASKTEACH will be used. These result in relatively simple data bases composed of lists of relevant features. As a part of the analysis, block diagrams at a suitable level of detail for illustrating BIT/AWG-9 topography will be produced. These will be the basis for interactive block diagrams implemented by the IMLAC computer system. Some appreciation for the complexity of these systems can be gained from referring to the Appendix.

Task Structures

The box labeled task analysis in figure 1 was used in the present case of the AWG-9 and BIT, to develop very thorough documentation with the aid of the different hardware contractors. This includes detailed analyses of tasks, in the form of lists of prescribed steps for maintenance procedures. These documents have been made available to BTL. Consequently, they are being used as the sources of information for task analysis.

The manual operating procedures for running BIT are thoroughly prescribed. These procedures will be simulated by the computer programs that will track the student while he is operating on the simulated man-machine interface.

The use of BIT is embedded in a broader context of task prescriptions, which do become quite lengthy, as in the case of fault-localization logic trees, and which do require the technician to refer to several different, bulky, publications while doing line maintenance. It is an objective of the trainer to teach students to use the standard Navy documentation that they would have to use in operating environments.

The Instructional Strategy

The instructional strategy used in PSO CAI, which is basically vertical iteration up and down levels of complexity, is only one approach to the management of instruction. Very powerful strategies for sequencing multiple strands have been extensively implemented by Chant and Atkinson (1973), and by Suppes and Morningstar (1970). The general problem might be characterized as how to sequence instructional information through the very limited human input information-processing channel to build up effective internal representational structures, i.e.,mental models, in long-term memory to support skilled performance.

The general features of the PSO CAI instructional strategy are illustrated in figure 8, which summarizes the major elements and their relationships in this strategy.

The section of this diagram labeled front-loading refers to the points for students to enter the instructional system. It usually is the case that students need preliminary instruction before they can start to practice performing job skills. They need to absorb information specifically relating to, and supporting practice in performing job skills. One function of the front-loading section is to do this. Another purpose of this section is to bring students "up-to-speed" with respect to subskills in which they are deficient. Generally speaking, the three sections of this diagram are coordinate with instructing, practicing, and testing functions in training. Of course, the instructional sequence may cycle through these functions over and over, and they may be scheduled in the sequence in different configurations, which are dynamic features not implied in the diagram.

The box labeled "instructional aids" represents instructional techniques which would be applied during practice to facilitate the development of proficiency. External feedback (knowledge-of-results), induction of mental imagery, and visual analogies of invisible processes, fall into this category.

According to the conception of job performance as the integrated sequencing of a collection of subskills, students will require a certain amount of practice before they can sustain this integrated performance. Until then, they will have to function at lower levels. The chain of boxes labeled subskill drills is intended to indicate these levels of incomplete integration. They would be entered via remedial loops. Some of the subskills involved might also be scheduled as front-loading, particularly if they are common deficiencies in the population of students.

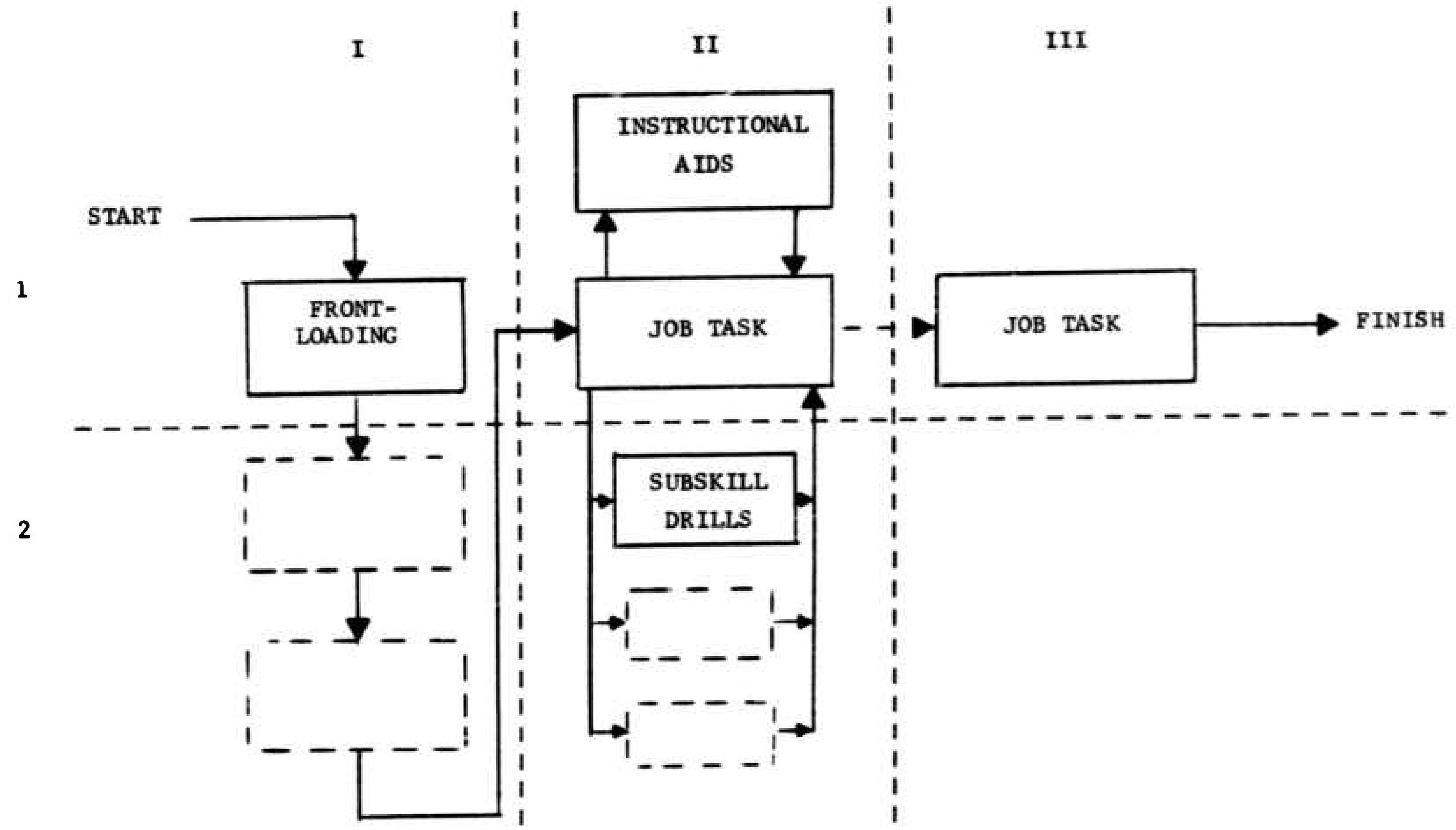


Figure 8. General Flowchart for PSO CAI Instructional Strategy

In Section II of the diagram, coordinate with the testing function instruction, the student's ability to sustain job task performance at criterion levels is established, and the student leaves the course. The RIO trainer is an example of the successful application of this general strategy (Rigney, et al., 1973b).

Instructional Strategy for the AWG-9 Trainer

The instructional strategy for the AWG-9 Maintenance Trainer will follow this general plan. The two major course elements that will be developed will be a simulation of operational BIT, called Opaque BIT Trainer (OPBITT) and Transparent BIT Trainer (TRANSBITT). The ways in which these elements will be interrelated is shown in figure 9.

In Stage I, the student will be given an overview of the trainer's functions. It will be assumed that the student will have been exposed at least to some of the introductory audio-visual materials used in classroom training, and will have some conception of what BIT is all about. Then, the student will run OPBITT to learn to operate each BIT sequence in the trainer and to see the elements of the final display for each BIT sequence. The student then will work through all the tests in the BIT sequence, using TRANSBITT, in which the displays will be interactive block diagrams. Time and error measures will be used by a student-monitor program to indicate when to transition the student into Stage II.

In Stage II, the student will begin to practice interpreting the OPBITT displays in terms of the test or tests that produced the information, and the fault area or fault areas in the AWG-9 from which the symptoms emanated. It will be necessary for the student to identify the basic parts of the BIT sequence display. (This is a performance requirement in the BQS manual). Then, the student will attempt to select the particular TRANSBITT block diagram at the finest-grained level of detail indicated by these diagrams. The student should be able to get into the right section of the top level diagram rather quickly. But he might require more practice before be could access the correct lowest-level diagram. Once at the correct lowest-level diagram, he could "rerun" the test to confirm that the malfunction was, in fact, in a particular fault area and that the test output was, in fact, a particular decision point (DP) number or set of DP numbers.

The student would proceed in this way, through a set of fault localization problems. Each problem would be created by simulating a failure in a particular fault area in a manner analogous to the logic used in TASKTEACH troubleshooting programs (Rigney, et al., 1972). Since OPBITT and TRANSBITT would use the same data base, the student could transfer from one to the other inside a problem. Time and error scores would be used to track the student's progress in learning to interpret OPBITT displays.

When the student met transitioning criteria, he would move into Stage III, where he would be tested on his proficiency in running BIT and in interpreting BIT displays.

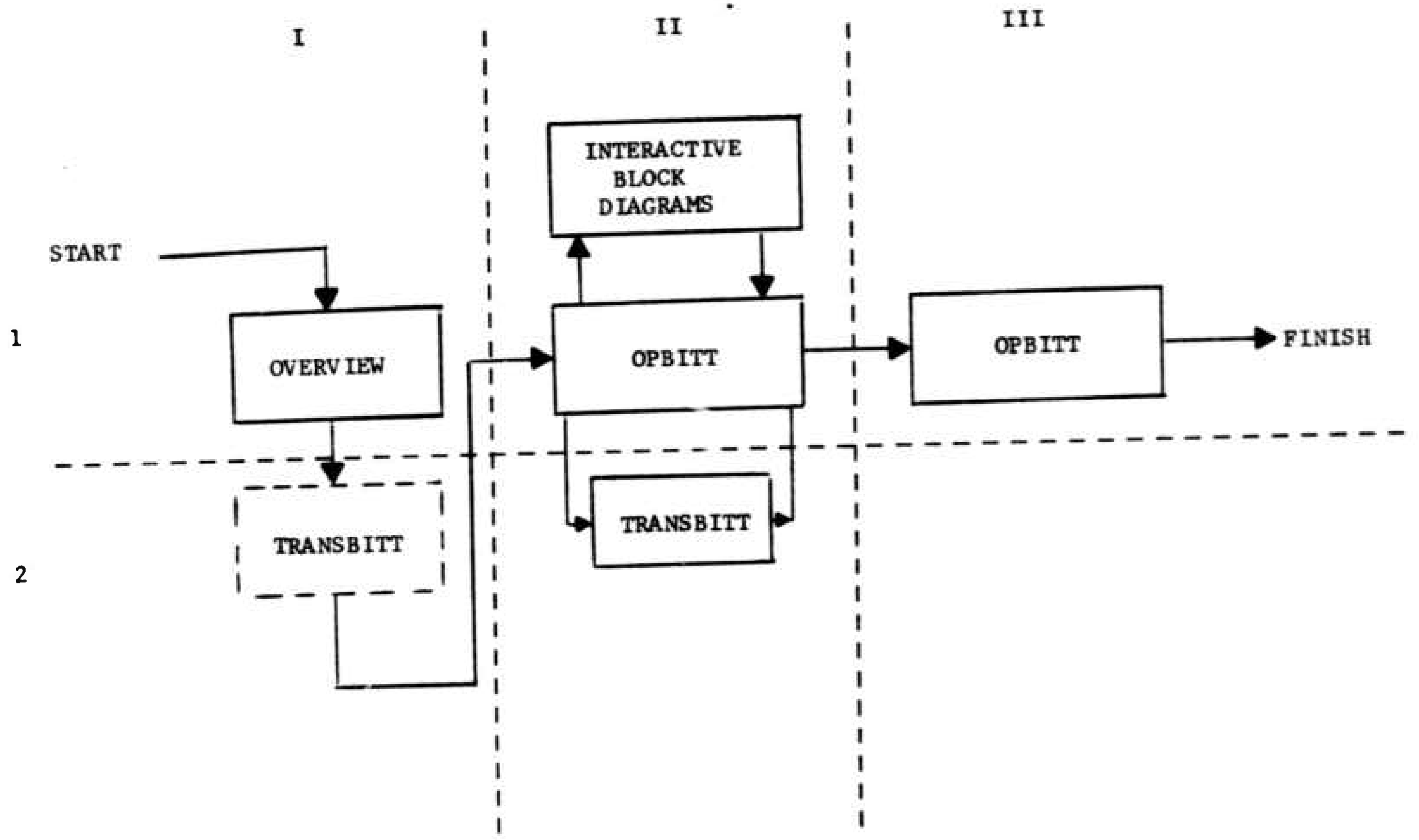


Figure 9. Top Level Instructional Strategy for AWG-9 Maintenance Trainer (Interactive Block Diagrams are the Instructional Aid used in TRANSBITT)

The organization illustrated in figure 9 will be used to guide this development. The single-thread approach to design of the trainer will result initially in data bases for two of the eight BIT sequences. The single-thread, or vertical-slice, approach is essential in early design stages to identify all of the types of data that will be required, so that program design can start, without becoming swamped by an attempt to develop an exhaustive data base from the beginning. An outline for an instructional sequence covering more topics than might be included in the trainer is presented, since some topics would be taught by the audio-visual techniques now being used by the Training Command.

In the following outline, "overview" is a combination of off-line techniques, such as sound/slide shows, and on-line drill. The on-line drill could involve the use of an on-line X-Y coordinate tablet and an on-line, random-access microfiche viewer. The tablet would accommodate schematic diagrams and other analogue representations in printed form, and would accept pointing responses from the student. Large amounts of technical information, in fact, all the technical manuals used in maintenance of the AWG-9, possibly excepting very large diagrams, could be put on microfiche, and access could be put under computer control. Various types of instruction could be provided in this way: prescriptions for maintenance actions, such as logic trees, or error-correcting feedback, or background information. The use of these devices will be considered in further planning.

An Instructional Sequence Outline

A. Basic relationships and characteristics of BIT.

1. Overview of AWG-9 and BIT.

2. a. Theory of operation of AWG-9.

   b. Theory of operation of BIT.

   c. BIT displays.

   d. Implicit logical structure within BIT structures.

   e. Use of documentation.

3. Operating Drill (OPBITT). The student interacts with CRT representations of the cockpit controls and displays to preset and to initiate BIT sequence 1-8.

4. Functional Drill (TRANSBITT). The student interacts with CRT representations of functional block diagrams to learn the fault areas (e.g., WRA's) monitored by each BIT sequence.

5. Augmented Performance. A combination of 2 and 3 - the student runs BIT, receives indications, and attempts to localize the fault area. The functional block diagrams serve as aids.

6. Performance Testing (OPBITT). Similar to 4 except that no augmentation is provided via functional block diagrams, and time and error scores are maintained.

B. Specific BIT Sequences: BIT Sequence 2*.

1. Overview of BIT Sequence 2 .

2. a. Theory of operation of AWG-9 functions monitored by BIT Seq. 2.

b. Theory of BIT sequence 2 (tests, inputs, outputs).

3. Operating Drill. The student interacts with CRT representations of the cockpit equipment to set up and initiate BIT Seq. 2.

4. Functional Drill. The student interacts with CRT representations of functional block diagrams to learn the fault areas and functional blocks monitored by BIT Seq. 2.

5. Augmented Performance. The student runs BIT Seq. 2, under a multitude of fault conditions, and attempts to localize the fault area to designated functions/blocks. He may call for replacement of WRA's and checking of bent connector pins. Faults introduced by the trainer at this stage do not require the use of BIT sequence digital entries.

6. Performance Testing. Same as A5.

C - H. Specific BIT Sequences: BIT Sequences 3-8; same instructional sequence as B.

I. Use of SPAM (scope to be defined later).

1. Overview of use of SPAM.

2. Operating Drill. The student interacts with CRT representations of the cockpit equipment to run the eight BIT sequences, followed by the appropriate SPAM procedures.

3. Functional Drill. The student interacts with CRT representations of functional block diagrams to learn the functions monitored by SPAM.

4. Augmented Performance. The student runs BIT, receives indications, and attempts to identify the faulty WRA and block. Problems simulated by the trainer in this stage require use of SPAM.

*It is recognized that BIT sequence 1 is atypical of subsequent BIT sequences, and that it would require special treatment if implemented in the trainer. For these reasons, implementation of this sequence in the trainer will be deferred. This outline applies more closely to BIT sequences 2-8.

5. Performance Testing (OPBITT). Similar to 4 except that no augmentation is provided via functional block diagrams, and time and error scores are maintained.

B. Specific BIT Sequences: BIT Sequence 2* .

1. Overview of BIT Sequence 2 .

2. a. Theory of operation of AWG-9 functions monitored by B̄T sequence 2.

   b. Theory of BIT sequence 2 (tests, inputs, outputs).

3. Operating Drill - The student interacts with CRT representations of the cockpit equipment to set-up and initiate BIT sequence 2.

4. Functional Drill - The student interacts with CRT representations of functional block diagrams to learn the fault areas and functional blocks monitored by BIT sequence 2.

5. Augmented Performance - The student runs BIT sequence 2, under a multitude of fault conditions, and attempts to localize the fault area to designated functions/blocks. He may call for replacement of WRA's and checking of bent connector pins. Faults introduced by the trainer at this stage do not require the use of BIT sequence digital entries.

6. Performance Testing - same as A5.

C - H. Specific BIT Sequences: BIT Sequences 3-8; same instructional sequence as B.

I. Use of SPAM (scope to be defined later).

1. Overview of use of SPAM.

2. Operating Drill - The student interacts with CRT representations of the cockpit equipment to run the eight BIT sequences, followed by the appropriate SPAM procedures.

3. Functional Drill - The student interacts with CRT representations of functional block diagrams to learn the functions monitored by SPAM.

4. Augmented Performance - The student runs BIT, receives indications, and attempts to identify the faulty WRA and block. Problems simulated by the trainer in this stage require use of SPAM.

*It is recognized that BIT sequence 1 is atypical of subsequent BIT sequences, and that it would require special treatment if implemented in the trainer. For these reasons, implementation of this sequence in the trainer will be deferred. This outline applies more closely to BIT sequences 2-8.

5. Performance Testing. Same as 4 but without error-correcting feedback.

J. Strategy of Applying the Complete BIT System to Maintenance of the AWG-9.

1. Overview of AWG-9 WCS testing and troubleshooting philosophy.

2. Operating Drill. The student interacts with a CRT representation of "WCS testing and troubleshooting philosophy" flowchart to learn the proper application of the BIT procedure.

3. Functional Drill. None at this level. If needed, return to A-J drills.

4. Augmented Performance. The student runs BIT, receives indications, and attempts to identify the fault area. He may call for replacement of WRA's, checking for bent pins, rerunning BIT sequences, and running SPAM. He will receive error-correcting feedback, based on standard documentation.

5. Performance Testing. Same as 4, but without feedback.

<u>Block Diagrams of Instructional Sequence</u>.

The instructional sequence is diagrammed below:

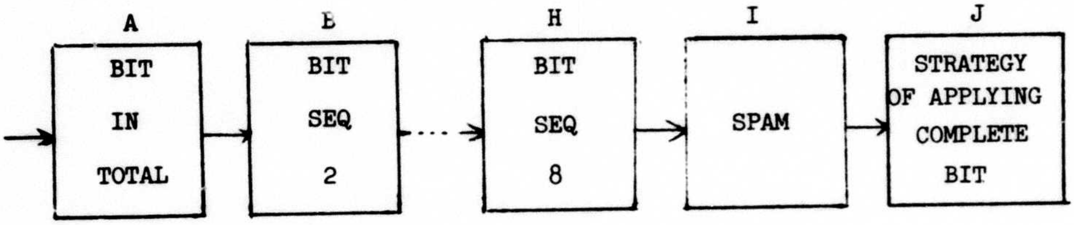


Figure 10. Subject-matter Modules for Macrointeractions

Each of the instructional modules A through J would contain a common sequence, as follows:

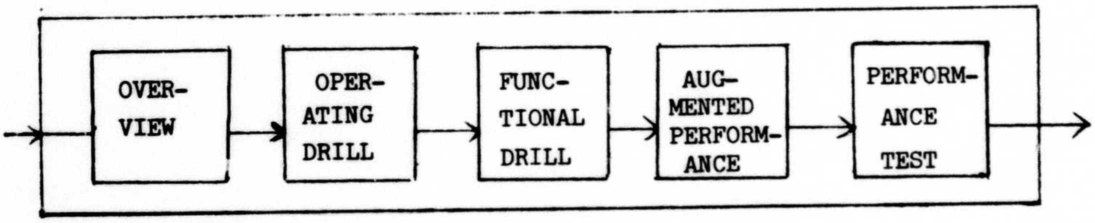


Figure 11. Sequence of Drills Within a Subject-matter Module

Software Implications Analysis

The generative approach to CAI (Carbonell, 1972; Rigney, et al., 1970) will be used. In this approach, the computer programs generate the interaction with the student from relatively simple data bases. In general terms, for this type of trainer, which requires simulation of job tasks and man/machine interfaces in the job environment, programs will be required for the following functions:

Man/Machine Interface Simulation--This will be done with techniques that have been developed at BTL (Rigney, et al., 1974b) for creating interactive computer graphics for simulating front panels and for displaying interactive block diagrams. Certain aspects of this type of program development have been greatly simplified by an interpreter which generates code for graphic displays while the operator is drawing these graphics; using function keys and the cursor on the graphics terminal.

Student-Program Interactions--While the analogue representational aspects of these will be supported by the interactive computer graphics, the digital representational aspects will require a series of routines to implement the sequence of drills described above, between and inside each block of subject-matter (figures 10 and 11). These programs will contain logic very similar to that in TASKTEACH troubleshooting programs for simpler equipment. However, the much greater complexity of the AWG-9/BIT systems, the fact that a BIT system is included and that the maintenance concept prescribes detailed maintenance procedures for the technician to follow, all dictate comprehensive additions to that logic. Consideration of requirements for sustaining the micro-interactions described above has led to the following general descriptions of program module functions. The discussion includes suggestions for sequencing criteria and for student data that might be used for performance histories for an adaptive controller (see figure 1).

Program Module Functions--The five basic program modules will now be characterized in more detail.

a. Overview. This would consist of several routines, all operating on the same data base. In the most general of characterizations, the data needed can be viewed as:

(1). A list of elements, each element possessing a list of properties.

For example:

Element I	Subband I	Element name
	5	BIT monitoring sequence
	110	Decision point
	039	WRA
	Doppler Processor	Name of sub-test in BIT sequence
	117.5	Page number in documentation

Subband II
Element II { .
.
.

(2). A group of interactive diagrams relating the elements in a meaningful way. (Preliminary versions of these diagrams are being prepared for BIT sequences 3 and 5).

The program then could execute numerous mapping and classification drills, as follows:

1. AWG-9 Organization. The trainer displays the element diagrams without WRA or other fault area names. In drill a the student points to the various blocks and the trainer displays the names. In drill b the trainer presents the names in random order and the student points out the blocks in the diagram. This is really a drill related to organization rather than names, since the student must associate a given name with an unlabeled block in a diagram.

2. BIT Sequence Organization. The trainer displays the diagrams, with WRA or other fault area names. In drill a the student points to the various names and the trainer presents the BIT sequences, and the student points out the appropriate blocks. This would include sequential dependencies across BIT sequences.

3. Functional Organization. This is similar to drill 1, except that the student is drilled in associating functional block names to unlabeled blocks within a WRA.

4. Test Theory. The trainer displays the diagrams with WRA or fault area names and functional block names. In drill a the student points to the names of a particular test (within a BIT sequence), and the trainer highlights the blocks monitored, the input point, and the point(s) of measurement. In drill b, the trainer presents the test names and the student indicates the blocks involved.

The program described above can exhibit great generality, since the property lists associated with elements can be quite flexible. Further, the general interaction is consistent; the program first performs the mapping of a particular characteristic onto the diagrams, then the roles are reversed and the trainer serves as a checker, supplying answers when the student is stumped.

Finally a simple set of parameters can be provided with the data base which governs the progression of the student through the various drills. Each mapping task (drill b) would have an associated time and error criterion which must be achieved before progressing to the next drill. Of course, the student would have greater control over rate and repetition of the a portions of each drill.

b. Operating Drill. The data base for the operating drill would consist of front-panel conditions, symptoms at indicators, various display lists (D-lists) which would produce "pictures" on the CRT, and, possibly, small task structures. Any operating drill can therefore be implemented by accessing a small list of pointers which specify where these data are stored on a storage device (disk). Upon loading these data (.02 - .5 seconds), the operating drill program would enter a "ready" loop, awaiting input from the student. Student inputs would consist primarily of control manipulations via a light pen. The trainer would simulate the operational equipment and record pertinent data regarding student performance. The operational drill can be considered a portion of Opaque BITT, which treats surface structures in the man/machine interface (figure 3). It would include digital entry operations.

c. Functional Drill. During the functional drill, the student interacts with functional block diagrams displayed on the CRT. The student practices identifying input/output points and functions tested for the subtests within a BIT sequence. The diagrams are dynamic in the sense that student inputs and program responses will affect them. These diagrams will serve to integrate information pertaining to BIT sequence test variables with fault areas and functional blocks in the AWG-9.

d. Augmented Performance. During augmented performance,the trainer simulates the observable aspects of the AWG-9 hardware and the BIT software. In addition, it injects malfunctions into the simulated environment. The student attempts to perform the maintenance task using the normally available documentation, but he may rely on the functional block diagrams provided in the functional drill. The student's objective is to become independent of these augmenting features, and perform exactly as he will when in the actual maintenance environment.

Research by others, and direct experience by this laboratory, indicates rather clearly that optional features under student-control are not effectively used by students. This may be because the material was not truly helpful. Or, perhaps the student is not in a position to decide if some material which he has not seen would be useful to him. In any case, the augmenting material here, the functional block diagrams, do not represent new information which was previously missing. Rather, the diagrams should serve to remind the student of relationships already covered during the functional drills, or perhaps to illustrate consequences of certain relationships in light of a particular malfunction situation. Thus, it appears that the diagrams might best be integrated into student performance by requiring the student to "explain" each problem in relation to the diagram. This might occur at the conclusion of each problem, or, if possible, would be done within a problem. The latter approach has the advantage of closer tracking of the student, allowing program intervention if the student embarks on an inappropriate activity.

In addition to using the diagrams to increase the fidelity of the trainer's evaluation of student performance, they also would be available for program or student control during each problem. Quite possibly the student also could return to certain portions of the functional drill if he, or the trainer, sensed a deficiency in some area. These options will be dealt with in the design of the macrointeraction loop (see figure 4).

e. Transitioning to Performance Testing. The only real difference between augmented performance and performance testing, in this planned trainer, is that the interactive block diagrams would not be provided. Thus,at some point the student must explicitly terminate the augmented performance stage and enter performance testing.

It is proposed that the trainer will determine when some prespecified criteria have been satisfied by the student during augmented performance, and will indicate that he may proceed to the performance testing stage. If desired, under certain conditions the student might continue in the augmented performance stage. He might continue until all problems have been worked or his allotted computer time has been expended.

The criteria for transitioning to performance testing will be variables which can be set by the Training Command. The trainer will maintain a number of performance measures from which the using facility can choose. Typical of these would be:

1. Number of problems attempted
2. Number of problems solved
3. Problems solved ÷ problems attempted
4. Average solution time per problem
5. Average number of WRA replacements per problem
6. Expected replacement time per problem (this is the time required to make the replacements requested by the student, as taken from flight line history)
7. Total expected repair time (#4 + #6 for each problem).

The transition rule might then be formulated as:

number of problems attempted	≥ 30
problems solved ÷ problems attempted	≥ .85
average solution time per problem	≤ 20 min.
expected replacement time per problem	≤ 1.7 hours.

Again, these are considerations for the design of the macrointeraction loop. When this loop is under program control, the adaptive controller, consisting of several programs, would do this controlling.

Adaptive Controller--Typical variables which might be included in student data were listed above. The point was made that criterion values of these should be adjustable by the Training Command. If this is desired, these variables also would be used for trials-to-criterion logic automatically implemented by a computer program. This approach was taken in the RIO trainer (Rigney et al., 1973b). This implementation requires several additional routines, for student data capture, data-analysis, and for monitoring student progress (see figure 2). Together, these routines would optimize instruction in terms of some goal, e.g., minimize time to complete the course, given fixed performance criteria, for a population of students. An important consequence of trials-to-criterion logic is that it results in reduced variance in student proficiency at the output end of the course.

Hardware Implications Analysis

The outstanding computer hardware requirement for CAI is storage. The IMLAC PDS-4 intelligent terminal will have 16K words of relatively fast-access core memory (.9 microseconds minimum) for running data-processing and display-processing programs. This will be augmented by floppy-disk storage of 256K words. Since these disks can be very quickly and simply inserted or removed from the disk controller, they also afford unlimited off-line storage.

Processor speeds are not considered to be a constraint in this self-standing system. Both the display processor and the central processing unit (CPU) are dedicated to one student at a time. With a display refresh-rate of 40 Hz, and a 900 nanosecond cycle time, there are very few occasions when display flicker would occur. An earlier version of this terminal with a 1.8 microsecond cycle time has been used extensively for a real-time performance training problem without experiencing any flicker.

The basic system; intelligent terminal with light pen, floppy-disk controller, and general purpose IO interface, will constitute the initial system. Several different peripheral devices are available as add-ons, controlled through the IO interface: a random-access slide projector, digital voice-synthesizer, microfiche projector, and a tablet. The microfiche projector would be attractive for storing the voluminous technical manuals associated with the AWG-9/BIT system, and for making the information readily accessible under computer control. Some consideration will be given to this possibility.

Scope of Development

The trainer to be developed will contain the program modules described earlier. These modules will be sufficiently general that they can function for any of the BIT sequences two through eight.

In addition they will function for the overall BIT stage (a), the testing and troubleshooting philosophy stage (j), and the memory inspection mode of SPAM (i). Initially, a data base for BIT sequence 5 and a data base for BIT sequence 3 will be prepared. The data for sequence 3 will be that required to transition to sequence 5.

The trainer, as initially put into the field, will include running two of the eight BIT sequences, performing digital entries for one BIT sequence, and the entire instructional strategy which applies to all BIT sequences.

At this point, consumer acceptance of the trainer will be tested, and consumer suggestions for modifications will be taken into consideration. Expansion of the trainer to teach BIT sequences 2, 6, 7, and 8 could be done by adding data modules for those sequences.

Since the preliminary specifications that have been described in this report are derived from a concept of trainers and of training that differs from the "big simulator" approach now commonly encountered, plans for testing of this particular trainer include, ultimately, cost-effectiveness comparisons with appropriate, currently existing examples of this approach. Performance criteria such as those listed on preceding pages will be used for evaluating training effectiveness. Estimating the cost of bringing students up to criterion levels of performance will require formulation of a multivariate cost-effectiveness model.

REFERENCES

Atkinson, R. C. and Paulson, J. A. An approach to the psychology of instruction. Psychological Bulletin, 1972, 78(1), 49-61.

Carbonell, Jaime. Scholar, A New Approach to Computer-assisted Instruction. Naval Research Reviews. October 1972. Arlington: Office of Naval Research, Department of the Navy.

Chant, V. G. and Atkinson, R. C. Optimal allocation instructional effort to interrelated learning strands. Journal of Mathematical Psychology, 1973, 10, 1-25.

Powers, W. T. Feedback: Beyond behaviorism. Science, 1973, 179, 351-356.

Rigney, J. W., Fromer, R. and Bond, N. A., Jr. The Application of Computer Time-sharing Technology to Training Requirements at the Electronics Schools Command, Mare Island. Los Angeles: Electronics Personnel Research Group, 1967 (TR 56).

Rigney, J. W. and Towne, D. M. Computer techniques for analyzing the microstructure of serial-action work in industry. Human Factors, 1969, 11(2), 113-122.

Rigney, J. W. and Towne, D. M. TASKTEACH: A method for computer-assisted performance training. Human Factors, 1970, 12(3), 285-296.

Rigney, J. W. Implications of Research on Internal Processing Operations in Learning and Memory for Serial Task Training. Los Angeles: Behavioral Technology Laboratories, 1971 (TR 67).

Rigney, J. W., Towne, D. M., King, C. A., and Langston, E. T. Computer-aided Performance Training for Diagnostic and Procedural Tasks. Los Angeles: Behavioral Technology Laboratories, 1972 (TR 70).

Rigney, J. W. A Discussion of Behavioral Technology Laboratories CAI Projects in Relation to a CAI Test-bed Concept. Los Angeles: Behavioral Technology Laboratories, 1973a (TR 71).

Rigney, J. W., Morrison, D. K., Williams, L. A. and Towne, D. M. Description and Initial Evaluation of a Computer-based Individual Trainer for the Radar Intercept Officer. Los Angeles: Behavioral Technology Laboratories (NAVTRAEQUIPCEN 71-C-0219-1), 1973b.

Rigney, J. W. Initial Considerations for Applying Performance-oriented CAI to F-14 Weapon System Maintenance Training. Los Angeles: Behavioral Technology Laboratoaies (NAVTRAEQUIPCEN/ARPA Quarterly Technical Report, April 15, 1974), 1974a.

Rigney, J. W., Towne, D. M., and King, C. A. Interactive Computer Graphics for Performance-Structure-Oriented CAI. Los Angeles: Behavioral Technology Laboratories, 1974b (TR 73).

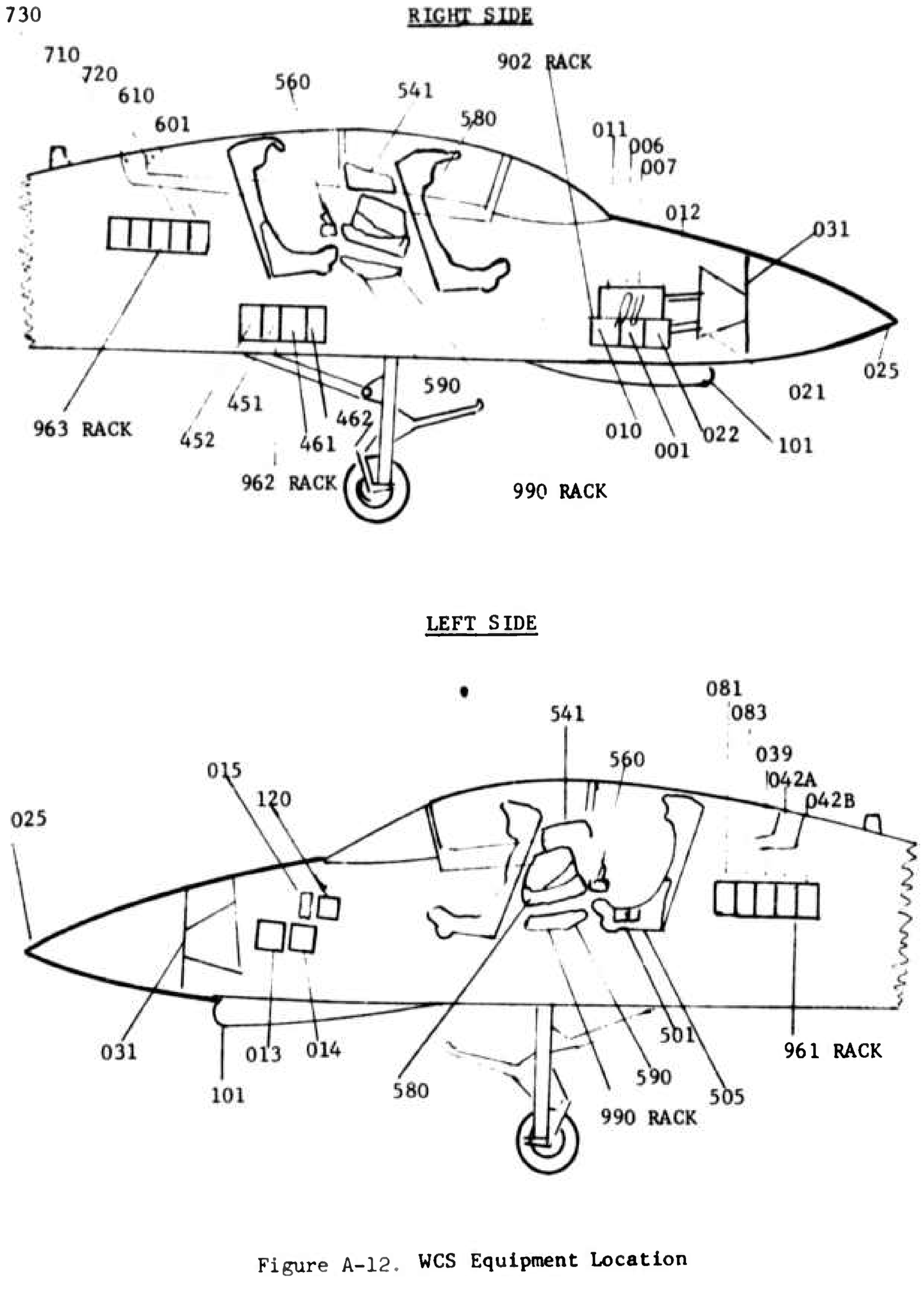
Rigney, J. A., Morrison, D. K., and Williams, L. A. A Guide for the Application of Performance-Structure-Oriented CAI in Naval Training Volume I. Los Angeles: Behavioral Technology Laboratories (NAVTRAEQUIPCEN 73-C-0065-1), 1974c.

Suppes, P. and Morningstar, M. Computer-assisted Instruction at Stanford, 1966-68: Data, Models, and Evaluation of the Arithmetic Programs. Stanford, Calif.: Institute for Mathematical Studies in the Social Sciences, 1972.

APPENDIX A

This is a brief overview of the BIT system in the AWG-9 WCS in the F-14. This material was abstracted from several Navy publications. All diagrams are from these publications. It is presented here merely for illustrative purposes.

The operation of BIT by the maintenance technician is performed from the rear cockpit of the aircraft, using tactical displays and controls which also serve for controlling BIT. The principal displays used for this purpose are the Tactical Information Display (TID), and the Detail Data Display (DDD). In addition, many controls have integral lights which turn on when the control is activated. The principal controls the maintenance technician used are grouped on panels: the Computer Address Panel, the TID panel, the Sensor Control Panel, the DDD panel, and the Hand Control Panel.


RIGHT SIDE

LEFT SIDE

Figure A-12. WCS Equipment Location

Rack numbers, WRA numbers, abbreviations, and names of the units shown in figure 12:

RIGHT SIDE

963 RACK:

730 - MAS - Missile power supply
710 - MAS - Missile data converter (MDC)
720 - MAS - Missile controller (MC)
601 - POWER - Semiregulated power supply (PWR)
610 - POWER - Regulated power supply (REG/PWR)
560 - C&D - Hand control (HC)
541 - DDD - Detail data display
580 - TID - Tactical information display

902 RACK:

011 - XMTR - Radar transmitter
006 - XMTR - Transmitter oscillator wave guide
007 - CWI - Continuous wave illuminator
012 - RADAR - Radar output wave guide (WG)
021 - RADAR - Radar input wave guide (WG)
031 - ANT - Radar antenna

---

025 - HORN - Radar test horn

---

101 - IR - Infrared receiver
022 - RCVR - Radar receiver
001 - RMO - Radar master oscillator
010 - RADAR - Radar synchronizer (RAD/SYNCH)

990 RACK:

590 - MREC - Mission recorder

962 RACK:

462 - PWR/TAPE - Power supply/bulk storer
461 - CPIO - Computer
451 - AC/DRO - Arithmetic and control/DRO
452 - NDRO - NDRO memory

LEFT SIDE:

015 - PWR - Solenoid power supply, radar (SOL/PWR)
120 - IR - Infrared amplifier
013 - PWR - Collector power supply (COL/PWR)
014 - PWR - Beam power supply (BEAM/PWR)

560 - C&D - Hand control
505 - C&D - Computer address (ADDRESS)
501 - C&D - Sensor control (SENS/CON)

081 - ANT/TC - Antenna and test controller
083 - RADAR - Low PRF processor (LOW PRF/PROC)
039 - RADAR - Doppler processor (DOPPL/PROC)
042A - DF - Doppler filter (DOPPL/FIL)
042B - DF - Doppler filter (DOPPL/FIL)

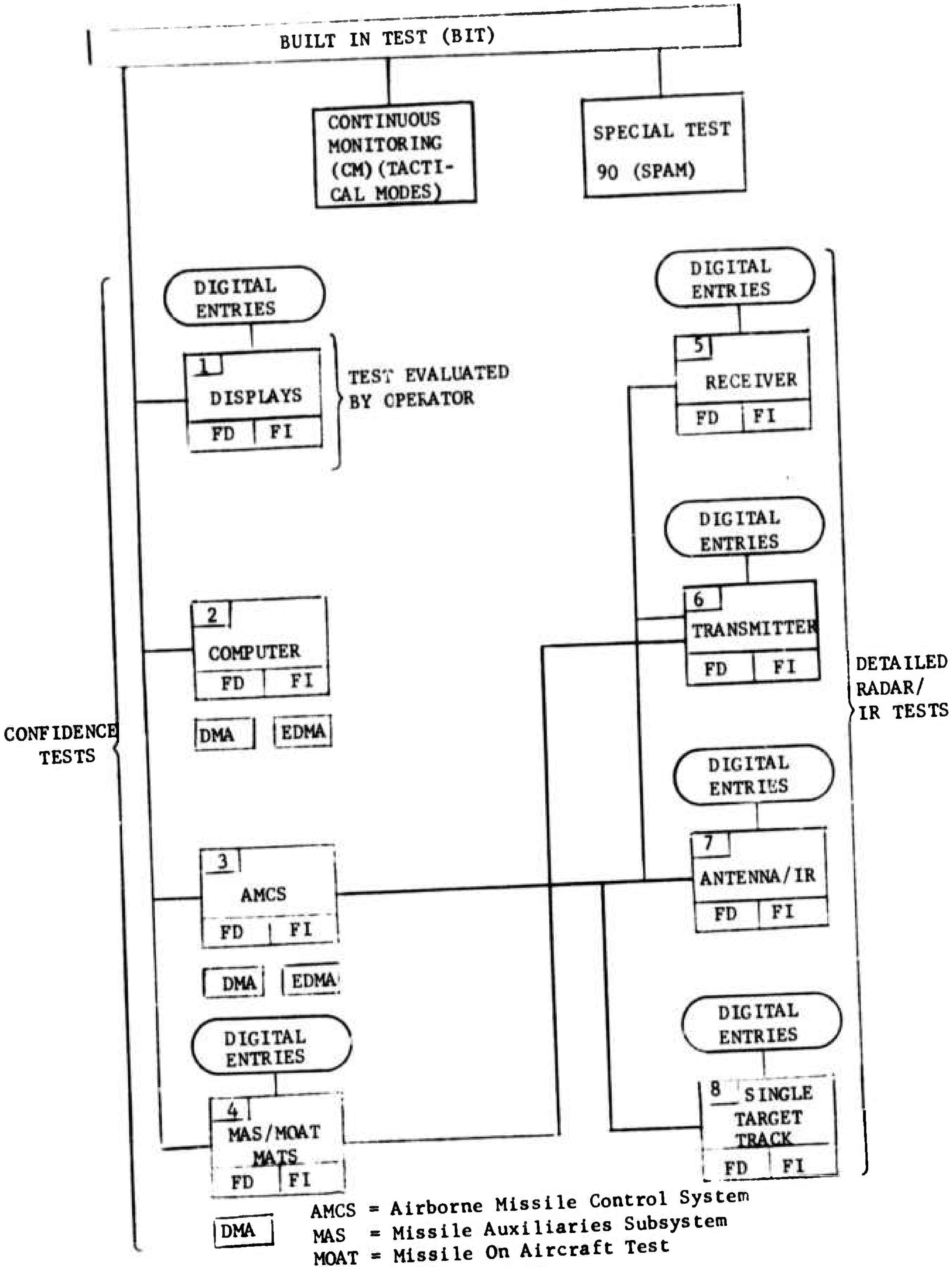
These rear cockpit controls and displays also are used for SPAM and FLYCATCHER. Manual operations and information displayed are different for these BEYOND BIT procedures.

The general organization of the BIT sequences is illustrated in figure 13. The continuous monitoring mode is used by the NFO (non-flying officer) during tactical procedures. Generally speaking, computer programs prepare the AWG-9 system for running BIT and run each BIT sequence. The AWG-9 system must be in the BIT state when SPAM tests are run. SPAM, Special Test 90, generally would be used following BIT, to perform more detailed analyses.

The first four BIT sequences are confidence tests. If they reveal no abnormal states, the remaining sequences are not run. BIT sequences five throught eight are detailed radar-IR tests that are run if BIT 3 or BIT 4 results indicate a malfunction is present in some possible fault area of the radar subsystems. The MAS/MOAT confidence test (BIT sequence 4) is not included in current planning for the trainer, since this is concerned with highly specialized missile systems.

The primary BIT control WRA's are 461, the central processor, and 081, the test controller logic housed with the antenna controller circuits. BIT sequence programs are stored on magnetic tapes in the bulk-storer housed in WRA 462. Some BIT sequences require intermediate core overlays.

There are, generally speaking, both analogue and digital interfaces between the CPU and the functional units under test (FUT's). The SSI LOOP is a digital part of this interface, SSI standing for Standard Serial Interface. Digital computer words are multiplexed and transmitted serially over this loop. Some outputs from the computer are called SOP's (serial output) and inputs to the computer are called SIP's (serial inputs). When a BIT program command results in an analogue output to a FUT, it is called an AO (analogue output). The returning analogue signal is called an AI (analogue input).



AMCS = Airborne Missile Control System
MAS = Missile Auxiliaries Subsystem
MOAT = Missile On Aircraft Test
FD - Fault Detection
FI = Fault Isolation
DMA = Degraded Mode Assessment (Basic)
EDMA = Expanded Degraded Mode Assessment

Figure A-13. BIT Organization

Each BIT sequence is composed of a series of tests. Each test is composed of manual test set-up (or initialization), which results in automatic injection of some type of signal (or using noise for a signal) at one point for the FUT, measurement of the output at another point for the FUT, comparison with a reference, and making the decision, "abnormal" or "normal." The points in the BIT sequence programs at which this decision is made are called decision points (DP's). These DP's are numbered. An abnormal test result also results in a DP number or numbers being stored in a buffer. At the end of running a BIT sequence, these DP numbers are collected and interpreted by the BIT program to generate the symptom information that is displayed for the technician on the TID in the rear cockpit.

It must be recognized that this is a highly simplified account. For example, a particular test in a BIT sequence may be run several times before a decision point is reached, and an entire BIT sequence may also be run several times in succession as a part of the maintenance procedures. Also, the exact kind of information that is displayed on the TID differs in important details from one BIT sequence to the next. One display that is common to BIT sequences 2-4 is the Degraded Mode Assessment (DMA). This assessment is cumulative from sequence 2 on, which is one reason why BIT sequences 2-4 must be run in exact sequence. The DMA display is illustrated in figure 14. The acronyms for the modes have the following meanings:

1. PDS - Pulse Doppler Search
2. RWS - Range While Search
3. TWS - Track While Scan
4. PDT - Pulse Doppler Single Target Track
5. BIT - Built In Test
6. PS - Pulse Search
7. AG - Air-to-Ground (PS with Antenna down)
8. PT - Pulse Single Target Track
9. ACM - Air Combat Maneuver
10. IRS - IR Scan (displayed if IR is on board)
11. TVS - TV Scan (displayed if TV is on board).

Special tests, e.g., SPAM and FLYCATCHER, probably are more subject to further development by the contractor as the weapon system matures. Consideration of these for the trainer will be deferred until later, with the exception of the SPAM memory inspection function. Some brief notes on SPAM follow:

SPAM is a contraction of SIPS, SOPS, PIPS, POPS, AOS, AIS, and MEMORY. It is identified as Special Test 90. SPAM is one type of test that would be used after running BIT. Where and when to use it is prescribed in the fault localization logic trees discussed in a previous section.

SPAM contains five subroutines that allow the operator to examine selected computer memory addresses, inputs, and outputs to aid in maintenance (only the memory inspection function would be implemented in the trainer in the near future):

1. NDRO (ron-destructive readout) memory stability test (run first: if NDRO is faulty, replace before proceeding with SPAM) (figure 15).

2. Serial input/serial output (SIP/SOP) subroutine.

3. Program input/program output (PIP/POP) subroutine: controls IFU (Interface Unit) functions.

4. Analogue input/analogue output (AI/AO) subroutine.

5. Memory inspection subroutine (figure 16).

It is emphasized that this is a very brief overview of the AWG-9 WCS and BIT, intended merely to acquaint the reader with some general features of the two systems. Full documentation for these systems occupies a number of large technical manuals. The contextual structure analysis for the development of data structures to describe essential features of the BIT sequences and tests requires an electronics engineer-senior programmer team to analyze each test within a BIT sequence in detail. The products of these analyses are data structures comprised of feature lists and block diagrams for BIT sequence tests.

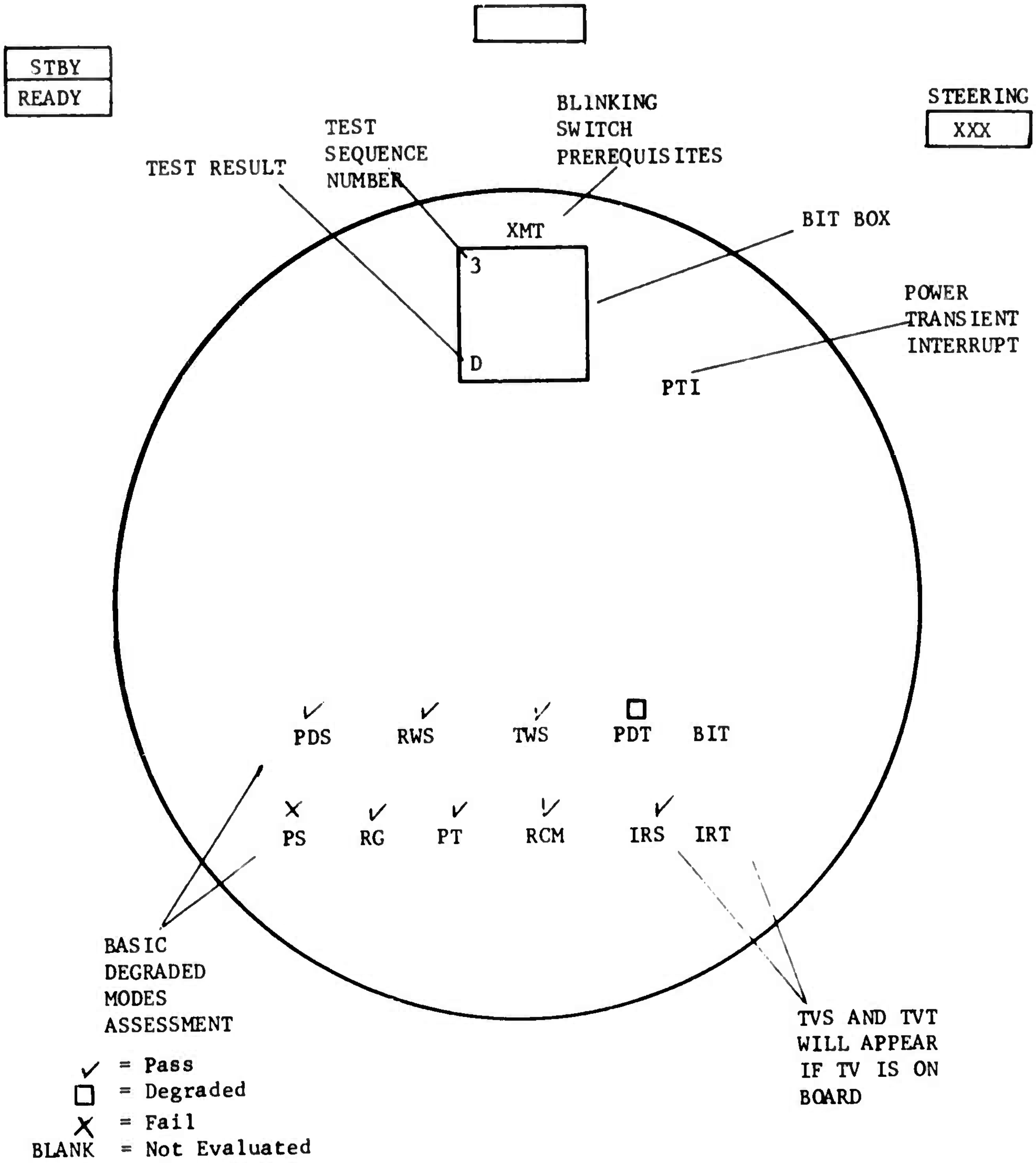


Figure A-14. Typical BIT Fault Detection Display

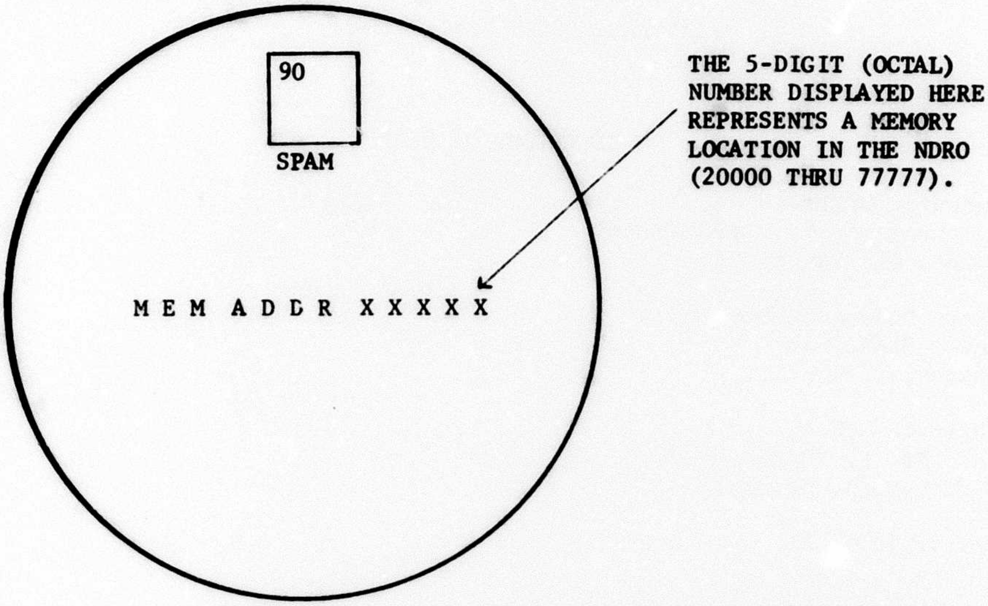


Figure A-15. NDRO Memory Stability Test Display

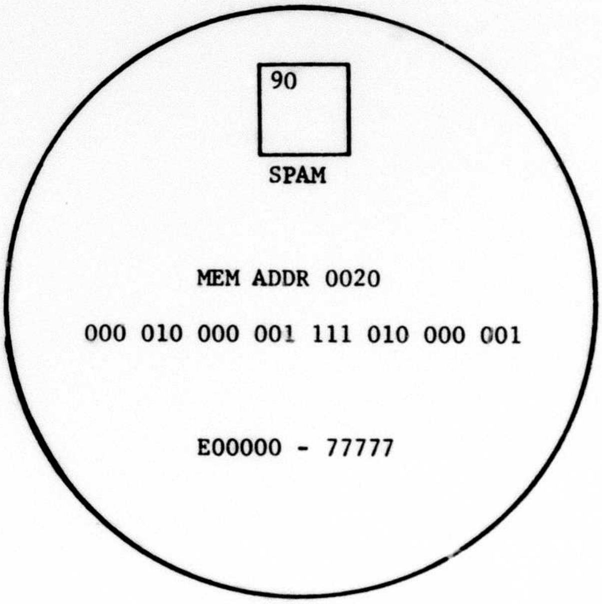


Figure A-16. Typical Special Test 90 Memory Inspection Routine Display